(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 February 2008 (28.02.2008)

PCT

(10) International Publication Number
WO 2008/023280 A2

(51) International Patent Classification: Not classified

(21) International Application Number: PCT/IB2007/003724

(22) International Filing Date: 12 June 2007 (12.06.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/804,546 12 June 2006 (12.06.2006) US

(71) Applicant *(for all designated States except US)*: **FOTONATION VISION LIMITED** [IE/IE]; Galway Business Park, Dangan, Galway City, Co. Galway (IE).

(72) **Inventors; and**
(75) **Inventors/Applicants** *(for US only)*: **IONITA, MIRCEA** [RO/IE]; Galway Business Park, Dangan, Galway (IE). **CORCORAN, PETER** [IE/IE]; Cregg, Claregalway, Co. Galway (IE).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

(54) **Title:** ADVANCES IN EXTENDING THE AAM TECHNIQUES FROM GRAYSCALE TO COLOR IMAGES

(57) **Abstract:** A face detection and/or detection method includes acquiring a digital color image. An active appearance model (AAM) is applied including an interchannel-decorrelated color space. One or more parameters of the model are matched to the image. Face detection results based on the matching and/or different results incorporating the face detection result are communicated.

ADVANCES IN EXTENDING THE AAM TECHNIQUES FROM GRAYSCALE TO COLOR IMAGES

PRIORITY

This application claims priority to United States provisional patent application no. 60/804,546, filed June 12, 2006, entitled “Improved Colour Model for Face Detection and Tracking” which is hereby incorporated by reference, and this application is related to United States Patent Application No. 11/761,647 filed June 12, 2007, and is hereby incorporated by reference

BACKGROUND

The active appearance model (AAM) techniques were first described by Edwards et al. [1]. They have been extensively used in applications such as face tracking and analysis and interpretation of medical images.

Different derivations of the standard AAM techniques have been proposed for grayscale images in order to improve the convergence accuracy or speed. Cootes et al. proposed in [2] a weighted edge representation of the image structure, claiming a more reliable and accurate fitting than using the standard representation based on normalized intensities. Other derivations include the direct appearance models (DAMs) [3], or the Shape AAMs [4], where the convergence speed is increased by reducing the number of parameters that need to be optimized. In the DAM approach, it is shown that predicting shape directly from texture can be possible when the two are sufficiently correlated. The Shape AAMs use the image residuals for driving the pose and shape parameters only, while the texture parameters are directly estimated by fitting to the current texture.

In [5], a method which uses canonical correlation analysis (CCAAM) for reducing the dimensionality of the original data instead of the common principal components analysis (PCA) is introduced. This method is claimed to be faster than the standard approach while recording almost equal final accuracy.

An inverse compositional approach is proposed in [6], where the texture warp is composed of incremental warps, instead of using the additive update of the parameters. This method considers shape and texture separately and is shown to increase the AAM fitting efficiency.

Originally designed for grayscale images, AAMs have been later extended to color images. Edwards et al. [7] first proposed a color AAM based on the RGB color space. This approach involves constructing a color texture vector by merging concatenated values of each color channel. However, their results did not indicate that benefits in accuracy could be achieved from the additional chromaticity data which were made available. Furthermore, the extra computation required to process these data suggested that color-based AAMs could not provide useful improvements over conventional grayscale AAMs.

Stegmann et al. [8] proposed a value, hue, edge map (VHE) representation of image structure. They used a transformation to HSV (hue, saturation, and value) color space from where they retained only the hue and value (intensity) components. They added to these an edge map component, obtained using numeric differential operators. A color texture vector was created as in [7], using instead of R, G, and B components the V, H, and E components. In their experiments they compared the convergence accuracy of the VHE model with the grayscale and RGB implementations. Here they obtained unexpected results indicating that the RGB model (as proposed in [7]) was slightly less accurate than the grayscale model. The VHE model outperformed both grayscale and RGB models but only by a modest amount; yet some applicability for the case of directional lighting changes was shown.

SUMMARY OF THE INVENTION

A method of detecting and/or tracking faces in a digital image is provided. The method includes acquiring a digital color image. An active appearance model (AAM) is applied including an interchannel-decorrelated color space. One or more parameters of the model are matched to the image. A face detection result based on the matching and/or a different processing result incorporating the face detection result is communicated.

The method may include converting RGB data to $I_1I_2I_3$ color space. The converting may include linear conversion. Texture may be represented with the $I_1I_2I_3$ color space. The texture may be aligned on separate channels. Operations may be performed on the texture data on each channel separately. The interchannel-decorrleated color space may include at least three channels including a luminance channel and two chromatic channels.

The AAM may include an application of principal components analysis (PCA) which may include eigen-analysis of dispersions of shape, texture and appearance. The AAM may further include an application of generalized procrustes analysis (GPA) including aligning shapes, a model of shape variability including an application of PCA on a set of shape vectors, a normalization of objects within the image with respect to shape and/or generation of a texture model including sampling intensity information from each shape-free image to form a set of texture vectors. The generation of the texture model may include normalization of the set of texture vectors and application of PCA on the normalized texture vectors. The applying may include retaining only the first one or two of the aligned texture vectors. The AAM may also include generation of a combined appearance model including a combined vector from weighted shape parameters concatenated to texture parameters, and application of PCA to the combined vector.

The matching may include a regression approach and/or finding model parameters and/or pose parameters which may include translation, scale and/or rotation.

The interchannel-decorrelated color space may include an orthogonal color space. Effects of global lighting and chrominance variations may be reduced with the AAM. One or more detected faces may be tracked through a series of two of more images.

An apparatus for detecting faces in a digital image is also provided including a processor and one or more processor-readable media for programming the processor to control the apparatus to perform any of the methods described herein.

BRIEF DESCRIPTION OF THE DRAWINGS

Figures 1a and 1b illustrate annotated images from PIE database and IMM database, respectively.

Figure 2 illustrates histograms of point-to-curve boundary errors after applying the (PIE) models on PIE Subset 1 (seen images).

Figure 3 illustrates cumulative histograms of pointtopoint boundary errors after applying the (PIE) models on PIE Subset 1 (seen images).

Figure 4 illustrates histograms of point-to-curve boundary errors after applying the (PIE) models on PIE Subset 2 (unseen images).

Figure 5 illustrates cumulative histograms of point-to-point boundary errors after applying the (PIE) models on PIE Subset 2 (unseen images).

Figure 6 shows actual dx vs. predicted dx displacements for (PIE) RGB GN model applied on PIE Subset 2.

Figure 7 shows actual dx vs. predicted dx displacements for (PIE) CIELAB GN model applied on PIE Subset 2.

Figure 8 shows actual dx vs. predicted dx displacements for (PIE) I1I2I3 SChN model applied on PIE Subset 2.

Figure 9 shows comparative average PtPt errors for PIE models applied on three different sets of images.

Figure 10 shows comparative average PtPt errors for IMM models applied on three different sets of images.

DETAILED DESCRIPTION OF THE EMBODIMENTS

A more appropriate extension of active appearance modeling (AAM) techniques to color images is provided. Accordingly, the embodiments are drawn to color spaces other than RGB because intensity and chromaticity information are strongly mixed in each of the R, G and B color channels. By employing color spaces where there is a stronger separation of the chromaticity and intensity information, we have been able to distinguish between intensity-dependent and chromaticity-dependent aspects of a color AAM. This has enabled the development of a new approach for normalizing color texture vectors, performing a set of independent normalizations on the texture subvectors corresponding to each color channel. This approach has enabled the creation of more accurate AAM color models than the conventional grayscale model. An investigation across a number of color spaces indicates that the best performing models are those created in a color space where the three color channels are optimally decorrelated. A performance comparison across the studied color spaces supports these conclusions.

The basic AAM algorithm for grayscale images is briefly described below. Then, extension of this model to RGB color images is analyzed, and a CIELAB-based model is proposed. CIELAB is a perceptually uniform color space that is widely used for advanced image processing applications. Extending the AAMs by applying the texture normalization separately to each component of the color space is also analyzed. The $I_1I_2I_3$ color space, which exhibits substantially optimal decorrelation between the color channels, is shown to be suited to this purpose. The proposed color AAM extension, which realizes a more appropriate texture normalization for color images is also described. Experimental results are shown, and a detailed set of comparisons between the standard grayscale model, the common RGB extension, and our proposed models are provided. Finally, conclusions are presented.

In what follows we frequently use the term texture. In the context of this work, texture is defined as the set of pixel intensities across an object, also subsequent to a suitable normalization.

OVERVIEW OF THE BASIC (GRAYSCALE) AAM

The image properties modeled by AAMs are shape and texture. The parameters of the model are estimated from an initial scene and subsequently used for synthesizing a parametric object image. In order to build a statistical model of the appearance of an object a training dataset is used to create (i) a shape model, (ii) a texture model and then (iii) a combined model of appearance by means of PCA, that is an eigenanalysis of the distributions of shape, texture and appearance. The training dataset contains object images, or image examples, annotated with a fixed set of landmark points. These are the training examples. The sets of 2D coordinates of the landmark points define the shapes inside the image frame. These shapes are aligned using the generalized Procrustes analysis (GPA) [9], a technique for removing the differences in translation, rotation and scale between the training set of shapes. This technique defines the shapes in the normalized frame. These aligned shapes are also called the shape examples.

Let N be the number of training examples. Each shape example is represented as a vector s of concatenated coordinates of its points $(x_1, x_2, ..., x_L, y_1, y_2, ..., y_L)^T$, where L is the number of landmark points. PCA is then applied to the set of aligned shape vectors reducing the initial dimensionality of the data. Shape variability is then linearly modeled as a base (mean) shape plus a linear combination of shape eigenvectors.

$$\mathbf{s}_m = \overline{\mathbf{s}} + \Phi_s \mathbf{b}_s, \quad (1)$$

where $\mathbf{s}_m$ represents a modeled shape, $\bar{s}$ is the mean of the aligned shapes, $\Phi_s = (\varphi_{s1} | \varphi_{s2} | ... | \varphi_{sp})$ is a matrix having p shape eigenvectors as its columns ($p < N$), and finally, $\mathbf{b}_s$ defines the set of parameters of the shape model. p is chosen so that a certain percentage of the total variance of the data is retained. The corresponding texture model is next constructed. For that, a reference shape is needed in order to acquire a set of so-called texture examples. The reference shape is usually chosen as the point-wise mean of the shape examples. The texture examples are defined in the normalized frame of the reference shape. Each image example is then warped (distorted) such that the points that define its attached shape (used as control points) match the reference shape; this is usually realized by means of a fast triangulation algorithm. Thus, the texture across each image object is mapped into its shape-normalized representation. All shape differences between the image examples are now removed. The resulting images are also called the image examples in the normalized frame. For each of these images, the corresponding pixel values across their common shape are scanned to form the texture vectors $\mathbf{t}_{im} =(\mathbf{t}_{im1} , \mathbf{t}_{im2} , ..., \mathbf{t}_{imP})^T$, where P is the number of texture samples.

Each texture vector $\mathbf{t}_{im}$ is further aligned with respect to intensity values, as detailed below, in order to minimize the global lighting variations. This global texture normalization is designed so that each normalized texture vector is aligned as closely as possible to the mean of the normalized texture vectors.

PCA is next applied to the set of normalized vectors, reducing thus the dimensionality of the texture data. The texture model is also a linear model, a texture instance being obtained from a base (mean) texture plus a linear combination of texture eigenvectors. Thus,

$$\mathbf{t}_m = \bar{\mathbf{t}} + \Phi_t \mathbf{b}_t. \tag{2}$$

Similar to the shape model, t_m represents a synthesized (modeled) texture in the normalized texture frame, $\bar{t}$ is the mean normalized texture, $\Phi_t = (\varphi_{t1} \, |\varphi_{t2}| \, ... |\varphi_{tq} \,)$ is a matrix having q texture eigenvectors as its columns, with q<N chosen so that a certain percentage from the total variance of the texture data is retained, and $\mathbf{b}_t$ defines the set of parameters of the texture model.

A vector **c** is further formed by concatenating the shape and texture parameters which optimally describe each of the training examples, $\begin{pmatrix} W_s b_s \\ b_t \end{pmatrix}$; $\mathbf{W}_s$ is a diagonal matrix of (normally equal) weights, applied in order to correct the differences in units between the shape and texture parameters.

A model for which the concatenated shape and texture parameters c are used to describe the appearance variability is called an independent model of appearance. A more compact model may be obtained by considering that some correlation exists between shape and texture. Thus, a third PCA is applied on the set of vectors **c**, resulting in a combined model of appearance

$$\mathbf{c}_m = \Phi_c \mathbf{b}_c, \tag{3}$$

where Φ_c is the matrix of retained eigenvectors and $\mathbf{b}_c$ represents the set of parameters that provide combined control of shape and texture variations. This reduces the total number of parameters of the appearance model.

During the optimization stage of an AAM (fitting the model to a query image), the parameters to be found are $p = \begin{pmatrix} g_s \\ b_c \end{pmatrix}$, where $\mathbf{g}_s$ are the shape 2D position, 2D rotation and scale parameters inside the image frame, and $\mathbf{b}_c$ are the combined model parameters.

The optimization of the parameters **p** is realized by minimizing the reconstruction error between the query image and the modeled image. The error is evaluated in the coordinate frame of the model, i.e., in the normalized texture reference frame, rather than in the coordinate frame of the image. This choice enables a fast approximation of a gradient descent optimization algorithm, described below. The difference between the query image and the modeled image is thus given by the difference between the normalized image texture and the normalized synthesized texture,

$$\mathbf{r(p)} = \mathbf{t} - \mathbf{t}_m, \quad (4)$$

and $||\mathbf{r(p)}||^2$ is the reconstruction error, with ||.|| marking the Euclidean norm.

A first order Taylor extension of **r(p)** is given by

$$\mathbf{r}(\mathbf{p}+\delta\mathbf{p}) \simeq \mathbf{r}(\mathbf{p}) + \frac{\partial \mathbf{r}}{\partial \mathbf{p}}\delta\mathbf{p}. \quad (5)$$

$\partial\mathbf{p}$ should be chosen so that to minimize $||\mathbf{r}(\mathbf{p}+\partial\mathbf{p})||^2$. It follows that

$$\frac{\partial \mathbf{r}}{\partial \mathbf{p}}\delta\mathbf{p} = -\mathbf{r}(\mathbf{p}). \quad (6)$$

Normally, the gradient matrix $\frac{\partial r}{\partial p}$ should be recomputed at each iteration. Yet, as the error is estimated in a normalized texture frame, this gradient matrix may be considered as fixed. This enables it to be pre-computed from the training dataset. Given a training image, each parameter in **p** is systematically displaced from its known optimal value retaining the normalized texture differences. The resulted matrices are then averaged over several displacement amounts and over several training images.

The update direction of the model parameters **p** is then given by

$$\delta\mathbf{p} = -\mathbf{R}\mathbf{r}(\mathbf{p}), \quad (7)$$

where $R = \left(\frac{\partial r}{\partial p}^{T} \frac{\partial r}{\partial p} \right)^{-1} \frac{\partial r}{\partial p}^{T}$ is the pseudo-inverse of the determined gradient matrix, which can be pre-computed as part of the training stage. The parameters **p** continue to be updated iteratively until the error can no longer be reduced and convergence is declared.

THE TEXTURE NORMALIZATION STAGE

As noted also by Batur *et al.* [10], and confirmed by our experiments, this stage is preferred during the optimization process, providing enhanced chances for predicting a correct update direction of the parameter vector ($\partial\mathbf{p}$).

Texture normalization is realized by applying to the texture vector t_{im} a scaling α, and an offset β, being thus a linear normalization,

$$\mathbf{t} = \frac{\mathbf{t}_{im} - \beta\mathbf{1}}{\alpha}, \qquad (8)$$

where 1 is the unity matrix.

The values for α and β are chosen to best match the current vector to the mean vector of the normalized data. In practice, the mean normalized texture vector is offset and scaled to have zero-mean and unit-variance. If $\frac{1}{N}\sum_{i=1}^{N} t_i$ is the mean vector of the normalized texture data, let $t_{zm,uv}$ be its zero-mean and unit-variance correspondent. Then, the values for α and β required to normalize a texture vector $\mathbf{t}_{im}$, according to (8), are given by

$$\alpha = \mathbf{t}_{im}^{T} \bar{\mathbf{t}}_{zm,uv}, \qquad (9)$$

$$\beta = \frac{\mathbf{t}_{im}^{T}\mathbf{1}}{P}. \qquad (10)$$

Obtaining the mean of the normalized data is thus a recursive process. A stable solution can be found by using one texture vector as the first estimate of the mean. Each texture vector is then aligned to zero mean and unit variance mean vector as described in (8)-(10), re-estimating the mean and iteratively repeating these steps until convergence is achieved.

COLOR AAM EXTENSIONS BASED ON GLOBAL COLOR TEXTURE NORMALIZATION

It is to be expected that using the complete color information will lead to an increased accuracy of AAM fitting. Yet, previous extensions of AAMs to color images showed only modest improvements, if any, in the convergence accuracy of the model. Before investigating this further, we first present the common AAM extension method to color images. We also propose a variant of this method based on a CIELAB color space representation instead of the initial RGB representation.

RGB is by far the most widely used color space in digital images [11]. The extension proposed by Edwards et al. [7] is realized by using an extended texture vector given by

$$\begin{aligned}\mathbf{t}_{im}^{RGB} = (\, & t_{im_1}^{R}, t_{im_2}^{R}, \ldots, t_{im_{P_c}}^{R}, \\ & t_{im_1}^{G}, t_{im_2}^{G}, \ldots, t_{im_{P_c}}^{G}, \\ & t_{im_1}^{B}, t_{im_2}^{B}, \ldots, t_{im_{P_c}}^{B} \,)^T,\end{aligned} \tag{11}$$

where P_c is the number of texture samples corresponding to one channel. Let $P = 3P_c$ denote now the number of elements of the full color texture vector.

In order to reduce the effects of global lighting variations, the same normalization method as for the grayscale model, described above, is applied on the full color texture vectors,

$$t_{im}^{RGB} \rightarrow t^{RGB} \quad (12)$$

The remaining steps of the basic grayscale algorithm remain unchanged.

CIELAB EXTENSION

CIELAB is a device-independent, perceptually linear color space which realizes a separation of color information into an intensity, or luminance component (L) and two chromaticity components (a, b). CIELAB was designed to mimic the human perception of the differences between colors. It is defined in terms of a transformation from CIE XYZ, which is a device-independent color space describing the average human observer. CIE XYZ is thus an intermediate space in the RGB to CIELAB conversion (RGB→XYZ→CIELAB); the detailed conversion steps are given in the Appendix.

The distance between two colors in the CIELAB color space is given by the Euclidean distance,

$$\Delta E = \sqrt{(\Delta L)^2 + (\Delta a)^2 + (\Delta b)^2} \quad (13)$$

CIELAB uses thus the same metric as RGB, and a CIELAB model implementation can be designed simply by substituting in (11) the values corresponding to the R, G, and B components with the values corresponding to the L, a, and b components, respectively. The color texture vector is thus built as

$$\begin{aligned}\mathbf{t}_{im}^{CIELAB} = (\ & t_{im_1}^{L}, t_{im_2}^{L}, \ldots, t_{im_{P_c}}^{L}, \\ & t_{im_1}^{a}, t_{im_2}^{a}, \ldots, t_{im_{P_c}}^{a}, \\ & t_{im_1}^{b}, t_{im_2}^{b}, \ldots, t_{im_{P_c}}^{b}\ )^T.\end{aligned} \quad (14)$$

Again, the same normalization technique can be applied on the resulted color vectors,

$$\mathbf{t}_{im}^{CIELAB} \longrightarrow \mathbf{t}^{CIELAB}. \quad (15)$$

The CIELAB AAM implementation is interesting as it offers the possibility for a more accurate image reconstruction, aimed towards a human observer. The benefits of this can clearly be noticed when the model is built using a specific image database and tested on another database with different image acquisition attributes (e.g. different illumination conditions). Considering that the image is typically represented in the more common RGB color space, the application of the CIELAB model may be realized at the expense of the added computational cost introduced by the conversion to CIELAB representation.

TEXTURE NORMALIZATION ON SEPARATE CHANNEL SUBVECTORS

When a typical multi-channel image is represented in a conventional color space such as RGB, there are correlations between its channels. Channel decorrelation refers to the reduction of the cross correlation between the components of a color image in a certain color space representation. In particular, the RGB color space presents very high inter-channel correlations.

For natural images, the cross-correlation coefficient between B and R channels is ≈0.78, between R and G channels is ≈0.98, and between G and B channels is ≈0.94 [12]. This implies that, in order to process the appearance of a set of pixels in a consistent way, one must process the color channels as a whole and it is not possible to independently analyze or normalize them.

This observation suggest an explanation as to why previous authors [7] obtained poor results being compelled to treat the RGB components as a single entity. Indeed, if one attempts to normalize individual image channels within a highly correlated color space such as RGB, the performance of the resulting model does not improve when compared with a global normalization applied across all image channels. In a preferred embodiment, however, each image channel is individually normalized when it is substantially decorrelated from the other image channels, and thus an improved color AAM is realized.

There are several color spaces which were specifically designed to separate color information into intensity and chromaticity components. However such a separation still does not necessarily guarantee that the image components are strongly decorrelated. There is though a particular color space which is desirable for substantially optimal image channel decorrelation.

A DECORRELATED COLOR SPACE

An interesting color space is $I_1I_2I_3$, proposed by Ohta et al. [13], which realizes a statistical minimization of the interchannel correlations (decorrelation of the RGB components) for natural images. The conversion from RGB to $I_1I_2I_3$ is given by the linear transformation in (16).

$$I_1 = \frac{R+G+B}{3}, \quad (16a)$$

$$I_2 = \frac{R-B}{2}, \quad (16b)$$

$$I_3 = \frac{2G-R-B}{4}. \quad (16c)$$

Similar to the CIELAB color space, I_1 stands as the achromatic (intensity) component, while I_2 and I_3 are the chromatic components. The numeric transformation from RGB to $I_1I_2I_3$ enables efficient transformation of datasets between these two color spaces.

$I_1I_2I_3$ was designed as an approximation for the Karhunen Loève transform (KLT) of the RGB data to be used for region segmentation on color images. The KLT is optimal in terms of energy compaction and mean squared error minimization for a truncated representation. Note that KLT is very similar to PCA. In a geometric interpretation, KLT can be viewed as a rotation of the coordinate system, while for PCA the rotation of the coordinate system is preceded by a shift of the origin to the mean point [14]. By applying KLT to a color image, it creates image basis vectors which are orthogonal, and it thus achieves complete decorrelation of the image channels. As the transformation to $I_1I_2I_3$ represents a good approximation of the KLT for a large set of natural images, the resulting color channels are almost completely decorrelated. The $I_1I_2I_3$ color space is thus useful for applying color image processing operations independently to each image channel.

In the previous work of Ohta et al., the discriminating power of 109 linear combinations of R, G, and B were tested on eight different color scenes. The selected linear combinations were gathered such that they could successfully be used for segmenting important (large area) regions of an image, based on a histogram threshold. It was found that 82 of the linear combinations had all positive weights, corresponding mainly to an intensity component which is best approximated by I_1. Another 22 showed opposite signs for the weights of R and B, representing the difference between the R and B components which are best approximated by I_2. Finally, the remaining 4 linear combinations could be approximated by I_3. Thus, it was shown that the I_1, I_2, and I_3 components in (16) are effective for discriminating between different regions and that they are significant in this order [13]. Based on the above figures, the percentage of color features which are well discriminated on the first, second, and third channel is around 76.15%, 20.18%, and 3.67%, respectively.

$I_1I_2I_3$BASED COLOR AAM

An advantage of this representation is that the texture alignment method used for grayscale models can now be applied independently to each channel. By considering the band subvectors individually, the alignment method described above can be independently applied to each of them as

$$(t^{I_1}_{im_1}, t^{I_1}_{im_2}, ..., t^{I_1}_{im_{P_c}}) \longrightarrow (t^{I_1}_1, t^{I_1}_2, ..., t^{I_1}_{P_c}), \qquad (17a)$$

$$(t^{I_2}_{im_1}, t^{I_2}_{im_2}, ..., t^{I_2}_{im_{P_c}}) \longrightarrow (t^{I_2}_1, t^{I_2}_2, ..., t^{I_2}_{P_c}), \qquad (17b)$$

$$(t^{I_3}_{im_1}, t^{I_3}_{im_2}, ..., t^{I_3}_{im_{P_c}}) \longrightarrow (t^{I_3}_1, t^{I_3}_2, ..., t^{I_3}_{P_c}). \qquad (17c)$$

The color texture vector is then rebuilt using the separately normalized components into the full normalized texture vector,

$$\begin{aligned}\mathbf{t}^{I_1I_2I_3} = (\ & t^{I_1}_1, t^{I_1}_2, ..., t^{I_1}_{P_c}, \\ & t^{I_2}_1, t^{I_2}_2, ..., t^{I_2}_{P_c}, \\ & t^{I_3}_1, t^{I_3}_2, ..., t^{I_3}_{P_c}\ )^T.\end{aligned} \qquad (18)$$

In this way, the effect of global lighting variation is reduced due to the normalization on the first channel which corresponds to an intensity component. Furthermore, the effect of some global chromaticity variation is reduced due to the normalization operations applied on the other two channels which correspond to the chromatic components. Thus, the AAM search algorithm becomes more robust to variations in lighting levels and color distributions.

This also addresses a further issue with AAMs which is their dependency on the initial training set of images. For example, if an annotated training set is prepared using a digital camera with a color gamut with extra emphasis on ”redness” (some manufacturers do customize their cameras according to market requirements), then the RGB-based AAM will perform poorly on images captured with a camera which has a normal color balance. A model, built using multi-channel normalization, is noticeably more tolerant to such variations in image color balance.

During the optimization process, the overall error function $||\mathbf{r}(\mathbf{p})||^2$ is replaced by the weighted error function $\sum_{i=1}^{3} w_i \, \| r(p) \|^2$. The set of weights that correspond to each color channel should be chosen so as to best describe the amount of information contained in that particular image channel. Evidently this is dependent on the current color space representation. For the $I_1I_2I_3$ color space, the percentages of color features found to be well discriminated for each channel were given above. Note that these percentages can also serve as estimates of the amount of information contained in each channel. Thus, they can provide a good choice for weighting the overall error function. The relative weighting of the error function may be used for texture normalization on separate channel sub-vectors.

As remarked also in [8], the common linear normalization applied on concatenated RGB bands as realized in the RGB implementation is less than optimal. An $I_1I_2I_3$ based model in accordance with certain embodiments herein uses a separate texture normalization method which is, as described below, a more suitable approach for color images.

Moreover, by employing the $I_1I_2I_3$ color space, a more efficient compaction of the color texture data is achieved. As the texture subvectors corresponding to I_1, I_2, and I_3 channels are significant in the order of ≈76%, ≈20%, and ≈4%, one can retain about 96% of the useful fitting information out of the first two texture sub-vectors only. Thus, a reduced I_1I_2 model can be designed with the performance comparable to a full $I_1I_2I_3$ model in terms of final convergence accuracy. Combined with the normalization method of separate texture subvectors in accordance with certain embodiments, a reduced I_1I_2 model is still more accurate than the original RGB model while the computational requirements are reduced by approximately one third.

A detailed discussion of results, summarized in Tables I to VI, now follows.

EXPERIMENTS

The performance of several models were analyzed in the color spaces discussed above. Both texture normalization techniques described were tested for face structure modeling. Use was made in tests of the appearance modeling environment FAME [15], modifying and extending it to accommodate the techniques described herein.

The convergence rates of AAMs are not specifically addressed herein. However, this work is envisioned to move towards real-time embodiments in embedded imaging applications.

The performance of the models is presented in terms of their final convergence accuracy. Several measures are used to describe the convergence accuracy of the models and their ability to synthesize the face. These are the point-to-point (PtPt) and point-to-curve (PtCrv) boundary errors, and the texture error. The boundary errors are measured between the exact shape in the image frame (obtained from the ground truth annotations) and the optimized model shape in the image frame. The point-to-point error is given by the Euclidian distance between the two shape vectors of concatenated x and y coordinates of the landmark points. The point-to-curve error is calculated as the Euclidian norm of the vector of distances from each landmark point of the exact shape to the closest point on the associated border of the optimized model shape in the image frame. The mean and standard deviation of PtPt and PtCrv are used to evaluate the boundary errors over a whole set of images. The texture error is computed as the Euclidian distance between the texture vector corresponding to the original image and the synthesized texture vector after texture de-normalization. This error is evaluated inside the CIELAB color space in order to have a qualitative differentiation between the synthesized images which is in accordance with the human perception. This is called the perceptual color texture error (PTE).

Two standard face image databases were used, namely the CMU PIE database [16] and the IMM database [17]. Color images of individuals with full frontal pose, neutral expression, no glasses, and diffuse light were used in these tests. Thus, a set of 66 images (640x486 pixels) was taken from the entire PIE database and a second set of 37 images (640x480 pixels) from the IMM database. These reduced sets of images are referred to below when mentioning the PIE and IMM databases. The images were manually annotated using 65 landmark points as shown in Fig. 1. Although the images in the IMM database were available with an attached set of annotations, it was decided to build an annotation set for reasons of consistency between the two image test sets.

For the PIE database, the first 40 images were used in building the models. The convergence accuracy was tested on the same set of 40 images, called PIE Subset 1 or seen data, and separately on the remaining 26 images, called PIE Subset 2 or unseen data. The IMM database was similarly split into IMM Subset 1, containing the first 20 images (seen data), and IMM Subset 2 with the remaining 17 images (unseen data). By doing this, how well the models are able to memorize a set of examples was analyzed, and also their capability to generalize to new examples. All models were built so as to retain 95% of the variation both for shape and texture, and again 95% of the combined (appearance) variation. For cross-validation, the PIE models were applied on the full IMM database, as well as the IMM models on the full PIE database.

The following AAM implementations were analyzed:

- standard grayscale model (Grayscale);
- RGB model with global color texture normalization (RGB GN);

and the proposed models,

- CIELAB model with global texture normalization (CIELAB GN);
- $I_1I_2I_3$ model with texture normalization on separate channel sub-vectors ($I_1I_2I_3$ SChN);
- I_1I_2 model with texture normalization on separate channel sub-vectors (I_1I_2 SChN);

and also the remaining (color space)/(normalization method) possibilities were added to provide a complete comparative analysis,

- RGB model with texture normalization on separate channel sub-vectors (RGB SChN);
- CIELAB model with texture normalization on separate channel sub-vectors (CIELAB SChN);
- $I_1I_2I_3$ model with global texture normalization ($I_1I_2I_3$GN);
- I_1I_2 model with global texture normalization (I_1I_2 GN).

The grayscale images were obtained from the RGB images by applying the following standard mix of RGB channels,

$$\text{Grayscale} = 0.30R + 0.59G + 0.11B. \quad (19)$$

The testing procedure for each model is as follows: each model is initialized using an offset for the centre of gravity of the shape of 20 pixels on the x coordinate and 10 pixels on the y coordinate from the optimal position in the query image. The optimization algorithm (see above) is applied, and the convergence accuracy is measured. Convergence is declared successful if the point-to-point boundary error is less than 10 pixels.

Fig. 2 and Fig. 4 present a histogram of PtCrv errors for landmark points on PIE database for the seen and unseen subsets, respectively. It can be observed that these errors are concentrated within lower values for the proposed models, showing improved convergence accuracy. As expected, better accuracy is obtained for the seen subset. Fig. 3 and Fig. 5 present the dependency of the declared convergence rate on the imposed threshold on PIE database for the seen and unseen data, respectively. This shows again the superiority of the proposed implementations.

In order to provide an indication on the relevancy of the chosen (-20,-10) pixels initial displacement, as well as to have an indication on the convergence range differences between the models, convergence accuracy was studied for a wider range of initial displacements on the x coordinate (dx), keeping the -10 pixels displacement on the y coordinate fixed. The tests were performed on PIE Subset 2 (unseen data) and are presented in Fig. 6 - Fig. 8 for the three main model implementations. The figures show diagrams of actual vs. predicted displacements on a range of -60 to 60 pixels from the optimum position. The predicted displacements are averaged with respect to all images in the analyzed dataset. The vertical segments represent one unit of standard deviation of each predicted displacement for the analyzed dataset of images. The converge range, given by the linear part of the diagram, is rather similar for multiple three model implementations. The RGB GN model seems to be able to converge for some larger displacements as well, yet the standard deviation of the predicted displacements rapidly increases with distance, which shows that the convergence accuracy is lost. On the other hand, although the CIELAB GN and $I_1I_2I_3$ SChN models have a more abrupt delimitation of their convergence range, they present a small and constant standard deviation inside their linear range, which shows a more consistent and accurate convergence. Also, the (20,10) pixels initial displacement, applied for all the other tests, is well inside the normal convergence range for any of the three models, which validates the choice made.

In Fig. 9 and Fig. 10 a comparative block diagram is presented of average PtPt errors on three different image datasets for the PIE models and IMM models, respectively. Note that these errors are consistently low (across all datasets) for the $I_1I_2I_3$ and the reduced I_1I_2 models with texture normalization on separate channel subvectors.

From Table I Table VI, the successful convergence rate for the three proposed models is consistently the best in comparison to all other model implementations, being usually much higher than for the grayscale model. An inconclusive result was obtained for IMM database (Table III and Table IV), where most of the studied models converged successfully on all images. Interestingly, it can be noticed that the RGB GN model does not outperform the grayscale model, the successful convergence rate being actually lower for some of the studied cases. In particular, for the cross-validation tests, when applying the PIE models on IMM database (Table V), the RGB GN model has a very poor rate, being actually outperformed by all other model implementations. For the same situation, all three proposed models have very high convergence rates, particularly the $I_1I_2I_3$ SChN model which registered a rate of 100%. Notable results were also obtained for the case of applying IMM models on PIE database (Table VI).

In terms of convergence accuracy (PtPt, PtCrv) and perceptual texture error, it can be seen that the CIELAB implementation is still dependent to some extent on the image acquisition conditions. This is caused by the limitation of the CIELAB implementation which cannot be efficiently used with texture normalization on separate channel sub-vectors. Some redundancy of RGB coordinates is removed by separating intensity and chromaticity data, yet the components are still coupled during texture normalization. Thus, although the results are improved over the RGB implementation for many of the tested image datasets, especially for the cross-validation tests (Table V and Table VI), they seem to lack consistency (see Table III and Table IV).

Much more consistent results were obtained for $I_1I_2I_3$ SChN and I_1I_2 SChN models, where the convergence accuracy is significantly improved over the RGB GN implementation for all studied datasets. For $I_1I_2I_3$ SChN model the perceptual texture error is also notably reduced for all datasets.

TABLE I
Convergence Results on (PIE) Subset 1 (*Seen*)

Model	Success	Pt-Crv	Pt-Pt	PTE
	[%]	(Mean/Std)	(Mean/Std)	(Mean/Std)
Grayscale	87.50	2.98/2.17	5.05/5.63	-
RGB GN	85.00	3.33/2.01	5.68/5.70	5.73/2.15
CIELAB GN	97.50	2.38/1.47	3.48/2.13	4.85/1.19
I1I2I3 SChN	**100**	**1.54/0.88**	**2.34/1.15**	**4.26/0.89**
I1I2 SChN	97.50	1.63/1.30	2.68/2.79	5.96/1.51
RGB SChN	90.00	2.54/2.54	4.78/6.89	5.20/2.47
CIELAB SChN	97.50	1.71/1.56	3.03/3.62	4.59/1.72
I1I2I3 GN	87.50	3.08/1.80	4.97/4.47	5.50/1.94
I1I2 GN	92.50	2.52/1.66	4.15/4.41	6.62/1.88

TABLE II
Convergence Results on (PIE) Subset 2 (*Unseen*)

Model	Success	Pt-Crv	Pt-Pt	PTE
	[%]	(Mean/Std)	(Mean/Std)	(Mean/Std)
Grayscale	88.46	3.93/2.00	6.91/5.45	-
RGB GN	80.77	3.75/1.77	7.09/4.99	7.20/2.25
CIELAB GN	**100**	2.70/0.93	4.36/1.63	5.91/1.19
I1I2I3 SChN	**100**	**2.60/0.93**	**4.20/1.45**	**5.87/1.20**
I1I2 SChN	96.15	2.76/1.11	4.70/2.31	6.95/1.37
RGB SChN	73.08	4.50/2.77	8.73/7.20	7.25/2.67
CIELAB SChN	88.46	3.51/2.91	6.70/8.29	6.28/2.09
I1I2I3 GN	92.31	3.23/1.21	5.55/2.72	6.58/1.62
I1I2 GN	88.46	3.30/1.37	5.84/3.55	7.49/1.70

TABLE III
Convergence Results on (IMM) Subset 1 (*Seen*)

Model	Success	Pt-Crv	Pt-Pt	PTE
	[%]	(Mean/Std)	(Mean/Std)	(Mean/Std)
Grayscale	**100**	1.19/0.37	1.70/0.38	-
RGB GN	**100**	0.87/0.19	1.30/0.29	2.22/0.51
CIELAB GN	**100**	1.36/0.72	1.99/1.09	2.63/1.02
I1I2I3 SChN	**100**	0.78/0.20	1.21/0.31	2.06/0.44
I1I2 SChN	**100**	**0.77/0.19**	**1.21/0.29**	11.88/2.31
RGB SChN	**100**	0.88/0.36	1.31/0.42	**2.02/0.44**
CIELAB SChN	95.00	1.49/2.03	3.30/7.68	2.99/2.28
I1I2I3 GN	**100**	1.19/0.57	1.71/0.80	2.49/0.87
I1I2 GN	**100**	1.09/0.44	1.61/0.67	12.00/2.27

TABLE IV
Convergence Results on (IMM) Subset 2 (*unseen*)

Model	Success	Pt-Crv	Pt-Pt	PTE
	[%]	(Mean/Std)	(Mean/Std)	(Mean/Std)
Grayscale	**100**	3.03/1.38	4.27/1.54	-
RGB GN	**100**	2.97/1.24	4.25/1.38	4.96/1.10
CIELAB GN	**100**	3.05/1.12	4.21/1.12	4.47/0.77
I1I2I3 SChN	**100**	**2.82/1.40**	**4.12/1.34**	4.43/0.80
I1I2 SChN	**100**	2.86/1.54	4.21/1.54	12.14/2.67
RGB SChN	**100**	2.88/1.17	4.20/1.38	**4.28/0.74**
CIELAB SChN	94.12	3.37/2.17	5.39/4.72	4.93/1.75
I1I2I3 GN	**100**	3.06/1.04	4.31/1.15	4.91/1.13
I1I2 GN	**100**	2.96/1.09	4.20/1.22	12.26/2.64

TABLE V

Convergence Results for PIE Models on IMM Db

Model	Success	Pt-Crv	Pt-Pt	PTE
	[%]	(Mean/Std)	(Mean/Std)	(Mean/Std)
Grayscale	21.62	9.13/3.76	24.26/14.36	-
RGB GN	5.41	9.27/1.77	19.99/4.86	11.68/1.57
CIELAB GN	94.59	4.00/1.02	6.69/1.85	9.92/0.94
I1I2I3 SChN	**100**	**3.73/0.94**	**5.55/1.22**	**6.07/1.14**
I1I2 SChN	94.59	4.69/1.40	7.10/2.08	12.89/2.29
RGB SChN	10.81	10.07/4.28	22.41/14.64	10.05/1.53
CIELAB SChN	48.65	8.78/4.72	20.37/18.11	8.94/3.04
I1I2I3 GN	59.46	5.17/1.56	10.84/5.07	10.24/1.31
I1I2 GN	51.35	5.35/1.65	11.96/5.24	15.11/2.20

TABLE VI

Convergence Results for IMM Models on PIE Db

Model	Success	Pt-Crv	Pt-Pt	PTE
	[%]	(Mean/Std)	(Mean/Std)	(Mean/Std)
Grayscale	36.36	6.90/3.33	16.07/10.70	-
RGB GN	36.36	7.18/2.82	15.73/7.83	17.06/3.15
CIELAB GN	**72.73**	5.83/2.31	**10.84/7.85**	10.35/2.61
I1I2I3 SChN	65.15	**5.52/3.24**	12.11/9.84	9.05/2.83
I1I2 SChN	56.06	6.07/3.47	13.87/11.42	9.98/2.73
RGB SChN	36.36	7.06/3.20	16.43/9.77	**8.64/2.32**
CIELAB SChN	13.64	8.62/2.49	21.16/7.98	9.62/2.22
I1I2I3 GN	34.85	7.65/3.05	18.02/12.14	12.84/3.09
I1I2 GN	25.76	8.83/4.74	26.35/31.15	11.65/3.39

DISCUSSION AND CONCLUSIONS

The embodiments described above have been analyzed with respect to how changes in color space representation of an image influence the convergence accuracy of AAMs. In particular, AAMs have been compared that have been built using RGB, CIELAB and $I_1I_2I_3$ color spaces. Both of the latter color spaces provide a more natural separation of intensity and chromaticity information than RGB. The $I_1I_2I_3$ color space also enables the application of more suitable color texture normalization and as a consequence model convergence is significantly improved.

From described experiments, it was deduced that it would make sense to normalize each color channel independently, rather than applying a global normalization across all three channels.

Thus, a more natural color texture normalization technique is proposed in certain embodiments, where each texture subvector corresponding to an individual color channel is normalized independently of the other channels. Although this approach cannot be successfully used with the common RGB representation, it was determined that some significant results can be achieved in color spaces where intensity and chromaticity information are better separated. In particular, it was found that the $I_1I_2I_3$ color space, which was specifically designed to minimize cross-correlation between the color channels, is an advantageously practical choice for this purpose.

Also, applying the same normalization as for grayscale images on an RGB color texture vector can occasionally lead to decreased convergence accuracy, as suggested in earlier research [8]. Thus, there is little rationale to use an RGB based model as the additional color data does not reliably improve model convergence and it will take three times as long to perform matching operations. For these reasons, the common RGB extension of the basic AAM is only interesting for the purpose of rendering the full color information.

Yet, by employing the $I_1I_2I_3$ color space coupled with texture normalization on separate channel subvectors, significant improvement in convergence accuracy is achieved as well as an accurate reconstruction of the current color image. The reconstruction accuracy, determined by analyzing the mean texture error, is also improved when compared with models based on other color spaces. By using the proposed $I_1I_2I_3$ model with texture normalization on separate channel subvectors, the optimization algorithm, which is typically based on a gradient descent approximation, is less susceptible to errors caused by local error function minima. Thus, the algorithm performance is also noticeably more robust.

More than 96% of relevant data is encapsulated in the I_1 and I_2 components of the $I_1I_2I_3$ color space. The difference between using an AAM derived from a full $I_1I_2I_3$ color space representation and one which is built by retaining only the first two channels is not very significant. Where the speed of convergence is most important, the reduced I_1I_2 model might be favored to a full $I_1I_2I_3$ model due to the lower dimensionality of the overall texture vector and the reduced computational requirements of this two-channel model.

The present invention is not limited to the embodiments described above herein, which may be amended or modified without departing from the scope of the present invention as set forth in the appended claims, and structural and functional equivalents thereof.

In methods that may be performed according to preferred embodiments herein and that may have been described above and/or claimed below, the operations have been described in selected typographical sequences. However, the sequences have been selected and so ordered for typographical convenience and are not intended to imply any particular order for performing the operations.

In addition, all references cited above herein, in addition to the background and summary of the invention sections, as well as US published patent applications nos. 2006/0204110, 2006/0204110, 2006/0098890, 2005/0068446, 2006/0039690, and 2006/0285754, and US patent applications nos. 11/464,083, 60/804,546, 11/554,539, 60/829,127, 60/773,714, 60/803,980, 60/821,956, and 60/821,165, which are to be or are assigned to the same assignee, are all hereby incorporated by reference into the detailed description of the preferred embodiments as disclosing alternative embodiments and components.

In addition, the following United States published patent applications are hereby incorporated by reference for all purposes including into the detailed description as disclosing alternative embodiments:

[1] G. J. Edwards, C. J. Taylor, and T. F. Cootes, “Interpreting face images using active appearance models,” in Proc. 3rd IEEE International Conference on Face & Gesture Recognition (FG ’98), 1998, pp. 300– 305.

[2] T. F. Cootes and C. J. Taylor, “On representing edge structure for model matching,” in Proc. IEEE Computer Vision and Pattern Recognition (CVPR’01), 2001, pp. 1114–1119.

[3] X. Hou, S. Z. Li, H. Zhang, and Q. Cheng, “Direct appearance models.” in IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2001, pp. 828–833.

[4] T. F. Cootes, G. J. Edwards, and C. J. Taylor, “A comparative evaluation of active appearance model algorithms.” in Proc. 9th British Machine Vison Conference. British Machine Vision Association, 1998, pp. 680– 689.

[5] R. Donner, M. Reiter, G. Langs, P. Peloschek, and H. Bischof, “Fast active appearance model search using canonical correlation analysis,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 10, pp. 1690–1694, 2006.

[6] I. Matthews and S. Baker, “Active appearance models revisited,” International Journal of Computer Vision, vol. 60, no. 2, pp. 135 – 164, November 2004, in Press.

[7] G. J. Edwards, T. F. Cootes, and C. J. Taylor, “Advances in active appearance models,” in International Conference on Computer Vision (ICCV’99), 1999, pp. 137–142.

[8] M. B. Stegmann and R. Larsen, “Multiband modelling of appearance,” Image and Vision Computing, vol. 21, no. 1, pp. 61–67, jan 2003. [Online]. Available: http://www2.imm.dtu.dk/pubdb/p.php?1421

[9] C. Goodall, “Procrustes methods in the statistical analysis of shape,” Journal of the Royal Statistical Society B, vol. 53, no. 2, pp. 285–339, 1991.

[10] A. U. Batur and M. H. Hayes, “Adaptive active appearance models.” IEEE Transactions on Image Processing, vol. 14, no. 11, pp. 1707– 1721, 2005.

[11] G. Sharma and H. J. Trussell, “Digital color imaging,” IEEE Transactions on Image Processing, vol. 6, no. 7, pp. 901–932, 1997. [Online]. Available: citeseer.ist.psu.edu/sharma97digital.html

[12] M. Tkalˇciˇc and J. F. Tasiˇc, “Colour spaces perceptual, historical and applicational background,” in IEEE, EUROCON, 2003.

[13] Y.Ohta, T. Kanade, and T. Sakai, “Color information for region segmentation,” in Computer Graphics and Image Processing, no. 13, 1980, pp. 222–241.

[14] J. J. Gerbrands, “On the relationships between SVD, KLT and PCA.” Pattern Recognition, vol. 14, no. 16, pp. 375–381, 1981.

[15] M. B. Stegmann, B. K. Ersbøll, and R. Larsen, “FAME a flexible appearance modelling environment,” IEEE Transactions on Medical Imaging, vol. 22, no. 10, pp. 1319–1331, 2003. [Online]. Available: http://www2.imm.dtu.dk/pubdb/p.php?1918

[16] T. Sim, S. Baker, and M. Bsat, “The CMU pose, illumination, and expression (PIE) database of human faces,” Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMURITR0102, January 2001.

[17] M. M. Nordstrøm, M. Larsen, J. Sierakowski, and M. B. Stegmann, “The IMM face database an annotated dataset of 240 face images,” Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK2800 Kgs. Lyngby, Tech. Rep., may 2004. [Online]. Available: http://www2.imm.dtu.dk/pubdb/p.php?3160

What is claimed is:

1. A method of detecting faces in a digital image, comprising:

(a) acquiring a digital color image;

(b) applying an active appearance model (AAM) including an interchannel-decorrelated color space;

(c) matching one or more parameters of the model to the image; and

(d) communicating a face detection result based on the matching or a different processing result incorporating said face detection result, or both.

2. The method of claim 1, further comprising converting RGB data to I1I2I3 color space.

3. The method of claim 2, wherein the converting comprises linear conversion.

4. The method of claim 2, further comprising representing texture with the I1I2I3 color space.

5. The method of claim 4, further comprising aligning the texture on separate channels.

6. The method of claim 4, further comprising performing operations on the texture data on each channel separately.

7. The method of claim 1, wherein said interchannel-decorrleated color space comprises at least three channels including a luminance channel and two chromatic channels.

8. The method of claim 1, wherein the AAM comprises an application of principal components analysis (PCA).

9. The method of claim 8, wherein said PCA comprises eigen-analysis of dispersions of shape, texture and appearance.

10. The method of claim 8, wherein the AAM further comprises an application of generalized procrustes analysis (GPA) including aligning shapes.

11. The method of claim 10, wherein the AAM further comprises a model of shape variability including an application of PCA on a set of shape vectors.

12. The method of claim 11, wherein the AAM further comprises a normalization of objects within the image with respect to shape.

13. The method of claim 12, wherein the AAM further comprises generation of a texture model including sampling intensity information from each shape-free image to form a set of texture vectors.

14. The method of claim 13, wherein the generation of the texture model comprising normalization of the set of texture vectors and application of PCA on the normalized texture vectors.

15. The method of claim 14, wherein the applying comprises retaining only the first one or two of the aligned texture vectors

16. The method of claim 14, wherein the AAM further comprises generation of a combined appearance model including a combined vector from weighted shape parameters concatenated to texture parameters, and application of PCA to the combined vector.

17. The method of claim 1, wherein the matching comprising a regression approach.

18. The method of claim 1, wherein the matching comprises finding model parameters or pose parameters or both.

19. The method of claim 18, wherein the pose parameters comprise translation, scale or rotation,

or combinations thereof.

20. The method of claim 1, wherein said interchannel-decorrelated color space comprises an orthogonal color space.

21. The method of claim 1, wherein effects of global lighting and chrominance variations are reduced with said AAM.

22. The method of claim 1, further comprising tracking one or more detected faces through a series of two of more images.

23. An apparatus for detecting faces in a digital image, comprising a processor and one or more processor-readable media for programming the processor to control the apparatus to perform a method comprising:

(a) acquiring a digital color image;

(b) applying an active appearance model (AAM) including an interchannel-decorrelated color space;

(c) matching one or more parameters of the model to the image; and

(d) communicating a face detection result based on the matching or a different result incorporating said face detection result, or both.

24. The apparatus of claim 23, wherein the method further comprises converting RGB data to I1I2I3 color space.

25. The apparatus of claim 24, wherein the converting comprises linear conversion.

26. The apparatus of claim 24, wherein the method further comprises representing texture with the I1I2I3 color space.

27. The apparatus of claim 26, wherein the method further comprises aligning the texture on

separate channels.

28. The apparatus of claim 26, wherein the method further comprises performing operations on the texture data on each channel separately.

29. The apparatus of claim 23, wherein said interchannel-decorrleated color space comprises at least three channels including a luminance channel and two chromatic channels.

30. The apparatus of claim 23, wherein the AAM comprises an application of principal components analysis (PCA).

31. The apparatus of claim 30, wherein said PCA comprises eigen-analysis of dispersions of shape, texture and appearance.

32. The apparatus of claim 30, wherein the AAM further comprises an application of generalized procrustes analysis (GPA) including aligning shapes.

33. The apparatus of claim 32, wherein the AAM further comprises a model of shape variability including an application of PCA on a set of shape vectors.

34. The apparatus of claim 33, wherein the AAM further comprises a normalization of objects within the image with respect to shape.

35. The apparatus of claim 34, wherein the AAM further comprises generation of a texture model including sampling intensity information from each shape-free image to form a set of texture vectors.

36. The apparatus of claim 35, wherein the generation of the texture model comprising normalization of the set of texture vectors and application of PCA on the normalized texture vectors.

37. The apparatus of claim 36, wherein the applying comprises retaining only the first one or two of the aligned texture vectors

38. The apparatus of claim 36, wherein the AAM further comprises generation of a combined appearance model including a combined vector from weighted shape parameters concatenated to texture parameters, and application of PCA to the combined vector.

39. The apparatus of claim 23, wherein the matching comprising a regression approach.

40. The apparatus of claim 23, wherein the matching comprises finding model parameters or pose parameters or both.

41. The apparatus of claim 40, wherein the pose parameters comprise translation, scale or rotation, or combinations thereof.

42. The apparatus of claim 23, wherein said interchannel-decorrelated color space comprises an orthogonal color space.

43. The apparatus of claim 23, wherein effects of global lighting and chrominance variations are reduced with said AAM.

44. The apparatus of claim 23, wherein the method further comprises tracking one or more detected faces through a series of two of more images.

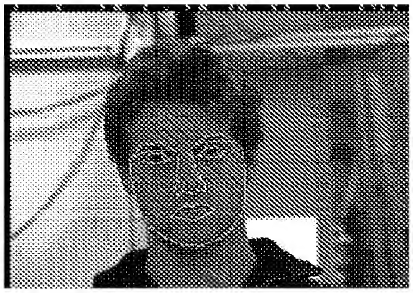

Figure 1a

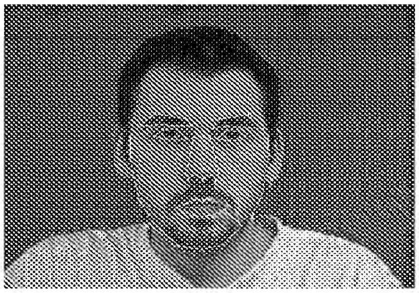

Figure 1b

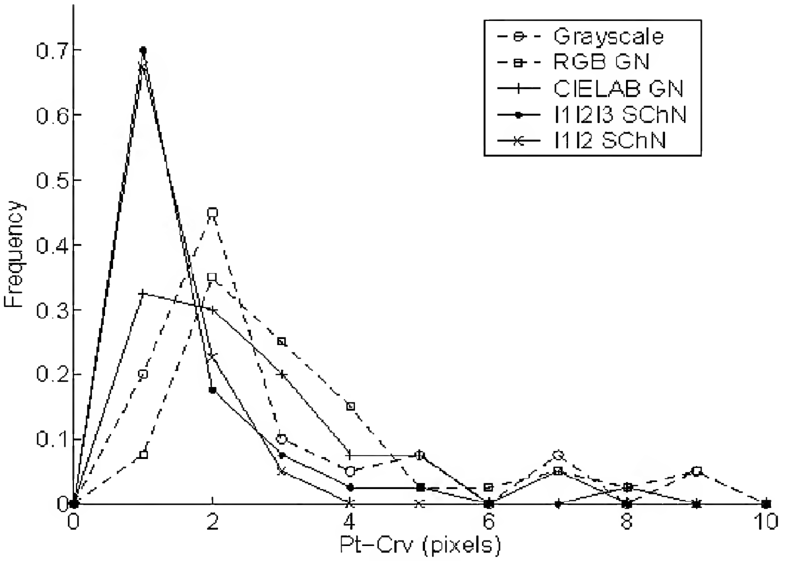

Figure 2

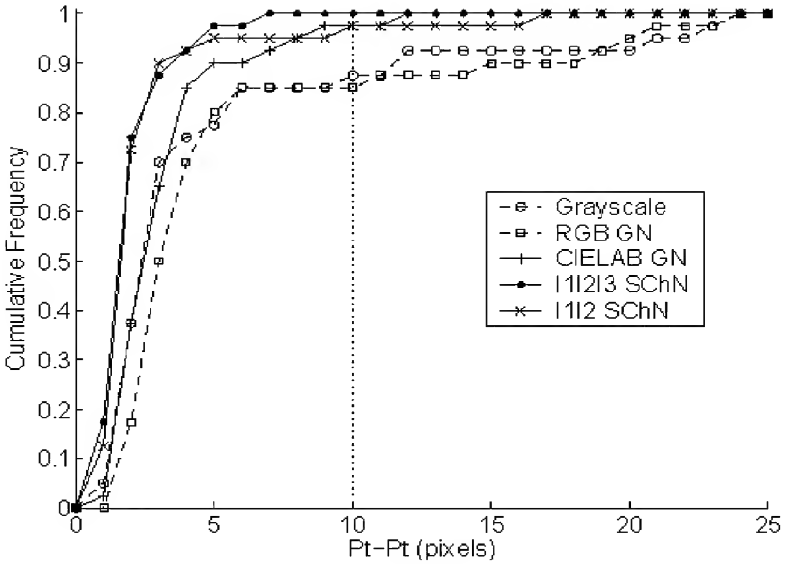

Figure 3

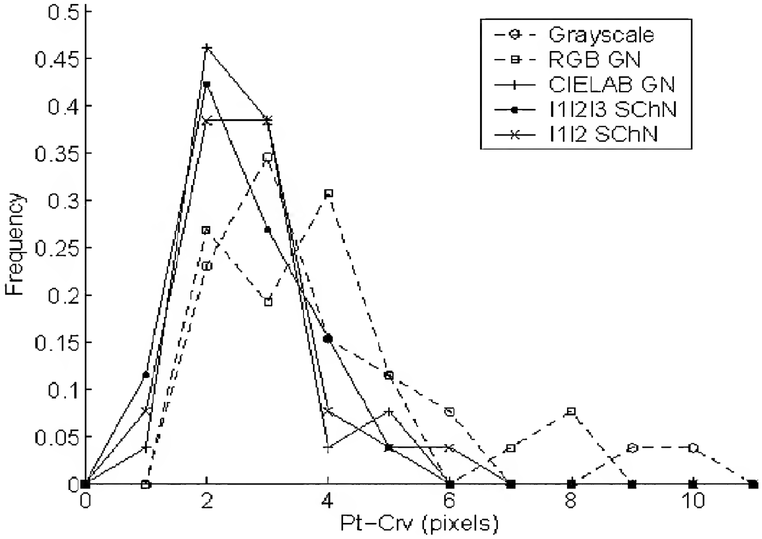

Figure 4

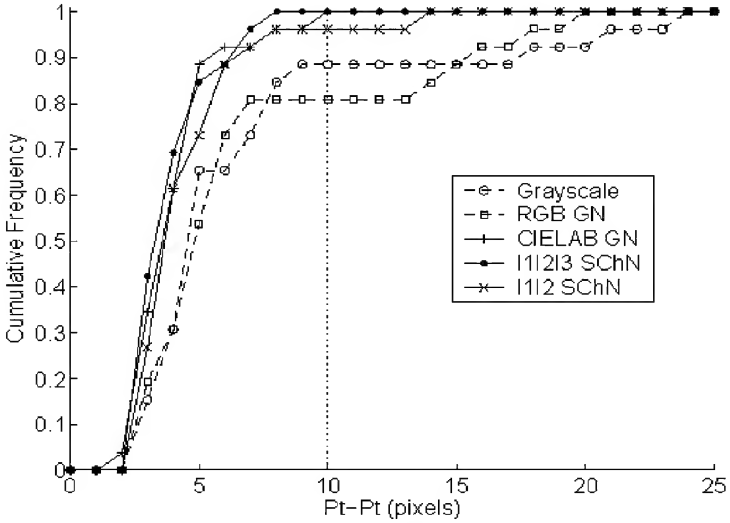

Figure 5

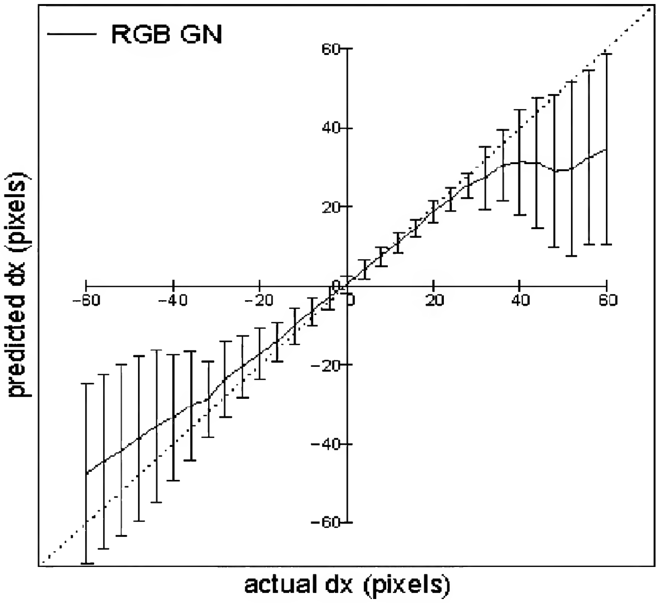

Figure 6

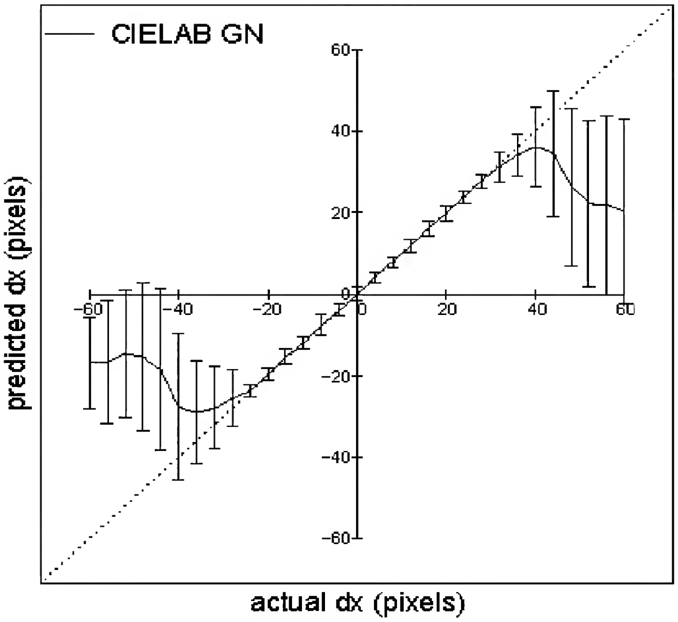

Figure 7

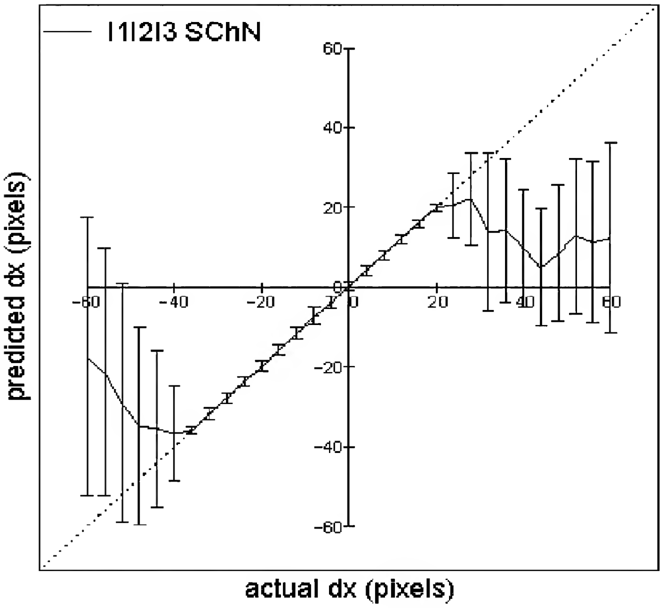

Figure 8

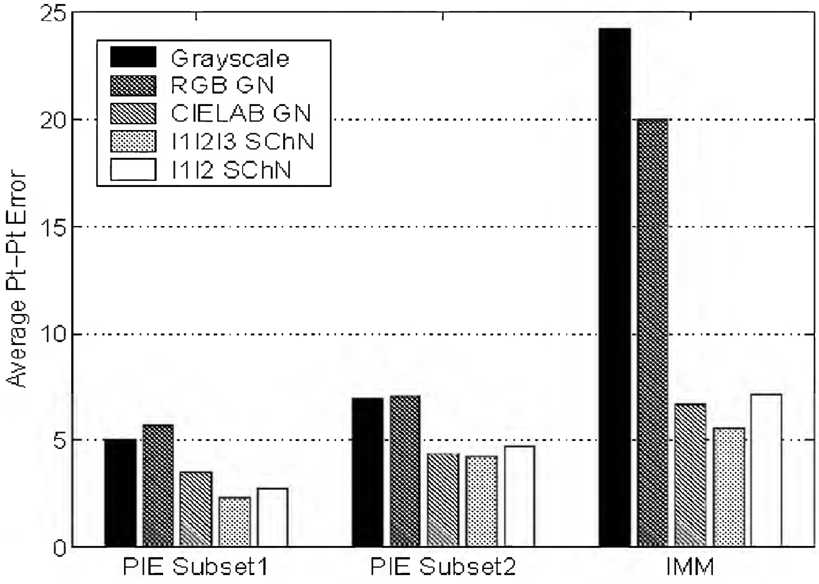

Figure 9

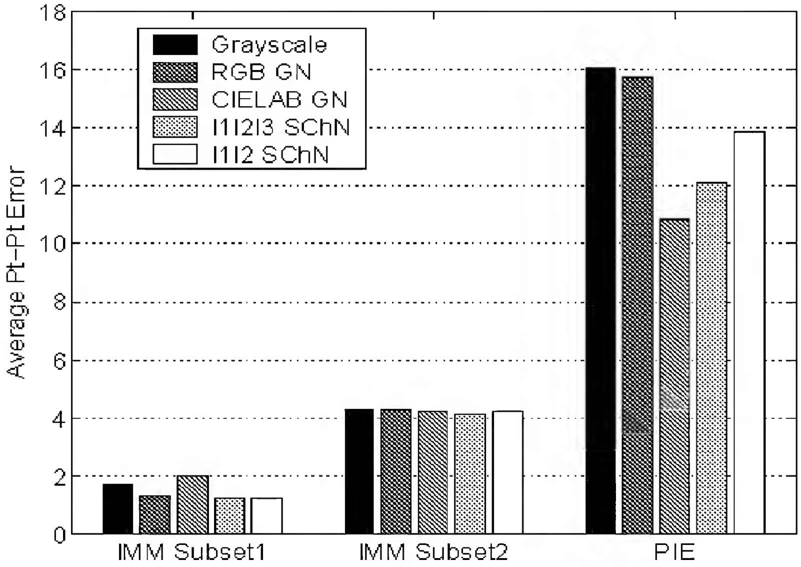

Figure 10